Journal of Computational Mathematics, Vol.25, No.5, 2007, 583–594.

FAST PARALLELIZABLE METHODS FOR COMPUTING INVARIANT SUBSPACES OF HERMITIAN MATRICES *

Zhenyue Zhang
(*Department of Mathematics, Zhejiang University, Hangzhou, China*
Email: zyzhang@zju.edu.cn)

Hongyuan Zha
(*College of Computing, Georgia Institute of Technology Atlanta, GA 30332, USA*
Email: zha@cc.gatech.edu)

Wenlong Ying
(*Department of Mathematics, Zhejiang University, Hangzhou, China*)

Abstract

We propose a *quadratically* convergent algorithm for computing the invariant subspaces of an Hermitian matrix. Each iteration of the algorithm consists of *one* matrix-matrix multiplication and *one* QR decomposition. We present an accurate convergence analysis of the algorithm without using the big O notation. We also propose a general framework based on implicit rational transformations which allows us to make connections with several existing algorithms and to derive classes of extensions to our basic algorithm with faster convergence rates. Several numerical examples are given which compare some aspects of the existing algorithms and the new algorithms.

Mathematics subject classification: 15A18, 65F05, 65F35.
Key words: Eigenvalue, Invariant subspace, Hermitian matrix, QR method, Parallelizable method.

1. Introduction

In [15] we proposed a cubically convergent algorithm for computing the two invariant subspaces of an Hermitian matrix A corresponding to the eigenvalues of A inside and outside the unit interval $[-1, 1]$, respectively. There we also presented a detailed convergence analysis which proved the cubic convergence of the algorithm. The derivation of the algorithm is inspired by the work in [1, 2, 3, 4, 6, 10, 11, 12] and the algorithm only uses matrix-matrix multiplications and QR decompositions as building blocks which are highly parallelizable primitive operations in libraries such as ScalaPack [14]. In this paper, we continue along the same line of research and concentrate on deriving new algorithms that can substantially reduce the amount of storage and the number of matrix-matrix multiplications. By exploiting the symmetry of the eigenvalue problem, we succeeded in deriving a new algorithm that employs only *one* matrix-matrix multiplication and *one* QR decomposition in each iteration. The presentation of the algorithm is the topic of Section 2. The structure of the new algorithm is extremely simple which allows us to give a much refined convergence analysis of the algorithm in Section 3. In particular, we were able to remove all of the big O expressions which were heavily used in [15]. The resulting bounds are cleaner and more concise. In Section 4, we analyze our proposed algorithm from the point of view of implicit rational transformations. This approach allows us to propose classes

* Received December 15, 2005; final revised September 1, 2006; accepted October 1, 2006.

of extensions of our basic algorithm which have higher convergence rates. To test the power of the implicit rational transformation framework, we will derive a simple version of the matrix sign function scheme from the general framework. We then discuss the relations of our new algorithms with Algorithm ISDA proposed in [1, 3, 10] and Algorithm CUBIC proposed in [15]. We focus on the accuracy of the invariant subspaces that are computed by those algorithms for a variety of numerical examples.

Remark. We want to emphasize that when the matrix A is non-Hermitian, then all the algorithms proposed in the sequel can be converted into algorithms for computing the singular subspaces of A.

2. The Algorithms

Our focus is to derive new algorithms which use as few matrix-matrix multiplications as possible in each iteration for computing an invariant subspace $\mathcal{V}_{(a,b)}$ of an Hermitian matrix $A \in \mathcal{C}^{n\times n}$ corresponding to the eigenvalues inside a preassigned interval $(a,\, b)$.[2)]

Theoretically, such an invariant subspace of A can be obtained by the following three steps. First, construct a function f that maps the complement of interval $[a,\, b]$ to zero and keep the image of $[a,\, b]$ far from zero. Second, compute the matrix function $f(A)$. Finally, compute the range space of $f(A)$ using QR algorithm column-pivoting to obtain the invariant subspace as required.

However, it is difficult to design such a function explicitly A feasible approach is to construct a function that has such properties approximately. We consider a sequence of functions $\{f_k\}$ that converges to an ideal f. One of the approaches for designing f_k is that we use a multiple composite of a fixed function g together with a scaling function ℓ,

$$f_k = \underbrace{g \circ \cdots \circ g}_{k \text{ times}} \circ \ell \equiv g^{(k)} \circ \ell \quad \text{with} \quad g^{(k)} = \underbrace{g \circ \cdots \circ g}_{k \text{ times}}, \tag{2.1}$$

where the iterative function g should be chosen such that 1) it has two invariant intervals I_1 and I_2 that cover the real space, i.e., $R = I_1 \cup I_2$, and 2) it shrinks one of the intervals, say I_1, as k increases, i.e., $g^{(k)}(I_1) \to \{\alpha\}$ as $k \to \infty$. The scaling function ℓ maps the inside of (a, b) into I_2 and the outside to I_1. This approach leads to an iterative method for computing the invariant subspace as follows.

Basic Iteration for Computing an Invariant Subspace.

1. Initial scaling. Set $B_0 = \ell(A)$.
2. Iteration. For $k = 0, \ldots,$ compute $B_{k+1} = g(B_k)$ until convergence.
3. Column-pivoting QR. Compute an orthogonal basis matrix of the range space of B_p for a convergent iterator B_p.

Obviously, $B_k = f_k(A)$ with f_k defined in (2.1). For ease of computation, the iterative function g should be chosen such that the matrix function $g(A)$ can be computed easily. In

[2)] We assume that a and b are not eigenvalues of A.

[10], a normalized incomplete Beta function β_i is used as the iterative function g.

$$\beta_i(x) = \frac{\int_0^x t^i(1-t)^i dt}{\int_0^1 t^i(1-t)^i dt} = \sum_{j=0}^{i} \binom{2i+1}{i-j}\binom{i+j}{j}(-1)^j x^{i+j+1},$$

where $\binom{k}{j} = \frac{k!}{j!(k-j)!}$. Indeed, only β_1 and β_2 are suggested, because the computational costs of $\beta_i(A)$ increase substantially for large i. A Beta function maps $[0, \frac{1}{2}) \to [0, \frac{1}{2})$, $(\frac{1}{2}, 1] \to (\frac{1}{2}, 1]$, and $\beta_i(\frac{1}{2}) = \frac{1}{2}$. Moreover,

$$\lim_{k\to\infty} \beta_i^{(k)}(x) \begin{cases} 0 & x \in [0, \frac{1}{2}) \\ 1 & x \in (\frac{1}{2}, 1] \end{cases}.$$

To determine a linear scaling function ℓ, an estimated interval (ω, Ω) is required to bound the eigenvalues $\lambda_j(A)$ of A, $\omega \le \lambda_j(A) \le \Omega$. ℓ is then chosen to be a linear function such that

$$\ell([\omega, \alpha]) \subset [0, \frac{1}{2}], \quad \ell([\alpha, \Omega]) \subset [\frac{1}{2}, 1]$$

for a given $\alpha \in (\omega, \Omega)$. Here we assume that α is not an eigenvalue of A. This method named as ISDA in [10] should be applied twice for computing the invariant subspace $\mathcal{V}_{(a,b)}$ corresponding to eigenvalues in the interval (a, b) if $(a, b) \subset (\omega, \Omega)$. We remark that ISDA can be applied for computing an invariant subspace of a real diagonalizable matrix [10] as well.

In this paper, we consider the following rational function as an iterative function g,

$$\phi(x) = \frac{x^2}{x^2 + (1-x)^2}. \tag{2.2}$$

Similar to β_i, ϕ has two invariant intervals $[0, \frac{1}{2})$ and $(\frac{1}{2}, 1]$. Note that $\frac{1}{2}$ is a fixed point of ϕ. Furthermore, $\phi^{(k)}([0, \frac{1}{2})) \to \{0\}$ and $\phi^{(k)}((\frac{1}{2}, 1]) \to \{1\}$ as $k \to \infty$. The scaling function ℓ can be a rational function such that it maps (a, b) into $(\frac{1}{2}, 1]$ and maps its complement set $[a, b]^c$ to $[0, \frac{1}{2})$. It is not difficult to verify that such an ℓ can be chosen as the following:

$$\ell(x) = \frac{c_1^2}{c_1^2 + (x - c_2)^2}, \quad c_1 = \frac{b-a}{2}, \quad c_2 = \frac{b+a}{2}.$$

Here ℓ and ϕ have similar representations. A question follows immediately: How can we compute the rational matrix function $\ell(A)$ or $\phi(B)$ for a given matrix A or B?

For an Hermitian matrix B, the matrix function $\phi(B)$ can be represented in the form

$$\phi(B)\big(B(B^2 + (I-B)^2)^{-1/2}\big)\big(B(B^2 + (I-B)^2)^{-1/2}\big)^H.$$

Notice that $B(B^2 + (I-B)^2)^{-1/2}$ is the top block of the normalized long matrix

$$\begin{bmatrix} B \\ I - B \end{bmatrix} (B^2 + (I-B)^2)^{-1/2}$$

that is an orthogonal basis matrix of the range space of $\left[\begin{smallmatrix} B \\ I-B \end{smallmatrix}\right]$. Obviously, it can be obtained by the QR decomposition of $\left[\begin{smallmatrix} B \\ I-B \end{smallmatrix}\right]$ within an orthogonal transformation. In practice, let $\left[\begin{smallmatrix} B \\ I-B \end{smallmatrix}\right] = QR = \left[\begin{smallmatrix} Q_1 \\ Q_2 \end{smallmatrix}\right] R$ be the QR decomposition. Then there is an orthogonal matrix M such that

$$\begin{bmatrix} B \\ I - B \end{bmatrix} (B^2 + (I-B)^2)^{-1/2} = QM = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} M,$$

It gives that $B(B^2 + (I - B)^2)^{-1/2} = Q_1 M$ and hence $\phi(B) = Q_1 Q_1^H$.

This observation motivates us to use QR decomposition for computing the matrix function $\phi(B)$ for an Hermitian matrix B. Similarly, let

$$\begin{bmatrix} c_1 I \\ A - c_2 I \end{bmatrix} = QR = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R$$

be the QR decomposition of $\left[\begin{smallmatrix} c_1 I \\ A - c_2 I \end{smallmatrix}\right]$. We have $\ell(A) = Q_1 Q_1^H$. The discussion above leads to the following algorithm for computing the invariant subspace $\mathcal{V}_{(a,b)}$ of A with eigenvalues in (a, b).

Algorithm QUAD.

1. *Initialization.* $B_0 = \frac{b-a}{2} I$, $Z_0 = A - \frac{a+b}{2} I$.
2. *Iteration.* For $k = 0, 1, 2, \ldots$, until convergence
   - 2.1 Compute QR decomposition $\begin{bmatrix} B_k \\ Z_k \end{bmatrix} \begin{bmatrix} Q_1^{(k)} \\ Q_2^{(k)} \end{bmatrix} R^{(k)}$.
   - 2.2 Update $B_{k+1} = Q_1^{(k)} (Q_1^{(k)})^H, \quad Z_{k+1} = I - B_{k+1}$.
   - 2.3 If $\|B_{k+1} - B_k\|_F \leq \texttt{tol}$, set $p = k + 1$ and terminate the iterations.
3. Compute the QR decomposition with column pivoting $B_p \Pi = QR$. The subspace spanned by the first r columns of Q is an approximate of $\mathcal{V}_{(a,b)}$ as required.

Here `tol` is a user supplied tolerance which will influence the accuracy of the computed approximate invariant subspace, r is the number of the largest diagonal entries of R far from the other diagonal elements. We have the following proposition that will be used in our convergence analysis proposed in the next section.

Proposition 2.1. *Denote $C = \frac{1}{c_1}(A - c_2 I)$. Using the notation in Algorithm* QUAD*, we have that for $k \geq 1$, $B_k = (I + C^{2^k})^{-1}$ and there are unitary matrices M_k such that*

$$Q_1^{(k)} = (I + C^{2^{k+1}})^{-1/2} M_k, \quad Q_2^{(k)} = C^{2^k} (I + C^{2^{k+1}})^{-1/2} M_k, \quad k = 0, 1, \ldots. \tag{2.3}$$

Proof. We prove the proposition using induction. Since both $\left[\begin{smallmatrix} Q_1^{(0)} \\ Q_2^{(0)} \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} I \\ C \end{smallmatrix}\right] (I + C^2)^{-1/2}$ are orthogonal basis matrices of the column space of $\left[\begin{smallmatrix} c_1 I \\ A - c_2 I \end{smallmatrix}\right]$, there exists a unitary matrix M_0 such that

$$\begin{bmatrix} Q_1^{(0)} \\ Q_2^{(0)} \end{bmatrix} = \begin{bmatrix} I \\ C \end{bmatrix} (I + C^2)^{-1/2} M_0.$$

Hence (2.3) holds for $k = 0$. By definition, $B_1 = Q_1^{(0)} (Q_1^{(0)})^H$.

In general, if $B_k = (I + C^{2^k})^{-1}$ for some $k = m \geq 1$, then $\left[\begin{smallmatrix} B_m \\ I - B_m \end{smallmatrix}\right]$ Therefore, both $\left[\begin{smallmatrix} Q_1^{(m)} \\ Q_2^{(m)} \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} I \\ C^{2^m} \end{smallmatrix}\right] (I + C^{2^{m+1}})^{-1/2}$ are orthogonal basis matrices of the column space of $\left[\begin{smallmatrix} B_m \\ I - B_m \end{smallmatrix}\right]$. It

follows that there is a unitary matrix M_m such that

$$\begin{bmatrix} Q_1^{(m)} \\ Q_2^{(m)} \end{bmatrix} = \begin{bmatrix} I \\ C^{2^m} \end{bmatrix} (I + C^{2^{m+1}})^{-1/2} M_m,$$

which implies $B_{m+1} = Q_1^{(m)}(Q_1^{(m)})^H = (I + C^{2^{m+1}})^{-1}$. By induction, the proposition holds.

3. Convergence Analysis

There are some interesting properties for the quantities in Algorithm QUAD. For example, B_k is an approximation of the *orthogonal projection* on to the invariant subspace $\mathcal{V}_{(a,b)}$ and M_k in Proposition 1 and $R^{(k)}$ approximate the identity matrix I. In this section we provide a detailed convergence analysis of Algorithm QUAD. All the quantities in Algorithm QUAD are carefully analyzed and rigorous bounds are given.

The eigenvalue decomposition of A is denoted by $A = Q\Lambda Q^H$, where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ with the first r eigenvalues inside (a, b) and the others outside $[a, b]$. We assume throughout the paper that a and b are not eigenvalues of A. Denote $\alpha(\lambda) = \frac{1}{c_1}(\lambda - c_2)$. Then $C = \alpha(A)$ and it has the eigen-decomposition $C = QDQ^H$ with

$$D = \mathrm{diag}(d_1, \ldots, d_n), \quad d_i = \alpha(\lambda_i), i = 1, \ldots, n.$$

Obviously, $|d_i| < 1$ for $i \le r$ and $|d_j| > 1$ if $j > r$. Partition $\Lambda = \mathrm{diag}(\Lambda_1, \Lambda_2)$ and $D = \mathrm{diag}(D_1, D_2)$ with $D_i = \alpha(\Lambda_i)$ if required. It gives $\|D_1\|_2 < 1$ and $\|D_2^{-1}\|_2 < 1$.

Theorem 3.1. *Let $P_{(a,b)}$ be the orthogonal projection onto the invariant subspace $\mathcal{V}_{(a,b)}$. If a and b are not eigenvalues of A, then*

$$\|B_k - P_{(a,b)}\| \le \eta^{2^k},$$

where $\eta = \max\left\{ |\alpha(\lambda_i)|, |\alpha(\lambda_j)|^{-1} \,\middle|\, \lambda_i \in (a,b),\ \lambda_j \notin [a,b] \right\} < 1$. *Moreover, if* $\eta^{2^k} \le 0.16$, *then*

$$\|B_{k+1} - B_k\| \le \|B_k - P_{(a,b)}\| < \|B_{k+1} - B_k\| + 2\|B_{k+1} - B_k\|^2.$$

Proof. By Proposition 2.1, $B_k = (I + C^{2^k})^{-1} = Q(I + D^{2^k})^{-1}Q^H$. It follows that

$$\begin{aligned} \|B_k - P_{(a,b)}\|_2 &= \|(I + D^{2^k})^{-1} - \mathrm{diag}(I_r, 0)\|_2 \\ &= \max\left\{ \frac{d_i^{2^k}}{1 + d_i^{2^k}}, \frac{d_j^{-2^k}}{1 + d_j^{-2^k}} \,\middle|\, i \le r < j \right\} \qquad (3.1) \\ &\le \max\{d_i^{2^k}, d_j^{-2^k} \mid i \le r < j\} = \eta^{2^k}. \end{aligned}$$

Furthermore, we have

$$\begin{aligned} \|B_{k+1} - B_k\|_2 &= \|(I + D^{2^{k+1}})^{-1} - (I + D^{2^k})^{-1}\|_2 \\ &= \max\left\{ \frac{d_i^{2^k}(1 - d_i^{2^k})}{(1 + d_i^{2^k})(1 + d_i^{2^{k+1}})}, \frac{d_j^{-2^k}(1 - d_j^{-2^k})}{(1 + d_j^{-2^k})(1 + d_j^{-2^{k+1}})} \,\middle|\, i \le r < j \right\}. \qquad (3.2) \end{aligned}$$

Comparing (3.1) and (3.2), we obtain that $\|B_{k+1} - B_k\|_2 \leq \|B_k - P_{(a,.b)}\|_2$. To show the upper bound of $\|B_k - P_{(a,.b)}\|_2$, let us denote $t = \frac{\epsilon(1-\epsilon)}{(1+\epsilon)(1+\epsilon^2)}$ and we write

$$\frac{\epsilon}{1+\epsilon} = t\frac{1+\epsilon^2}{1-\epsilon} = t(1 + \frac{\epsilon(1+\epsilon)}{1-\epsilon}) = t\left(1 + t\left(\frac{1+\epsilon}{1-\epsilon}\right)^2(1+\epsilon^2)\right).$$

It is not difficult to verify that for $\epsilon \in [0, 0.16]$, $\left(\frac{1+\epsilon}{1-\epsilon}\right)^2(1+\epsilon^2) < 2$. Thus $\frac{\epsilon}{1+\epsilon} < t(1+2t)$. Setting $\epsilon = \lambda_i^{2^k}$ and $\epsilon = (\lambda_j^{-1})^{2^k}$ for $i \leq r < j$, respectively, and comparing (3.1) and (3.2) again yields

$$\|B_k - P_{(a,b)}\|_2 < \|B_{k+1} - B_k\|_2(1 + 2\|B_{k+1} - B_k\|_2),$$

which completes the proof.

The above theorem gives a rigorous upper and lower bound for the approximation error $\|B_k - P_{(a,b)}\|_2$. It justifies the use of $\|B_{k+1} - B_k\|_2$ as the stopping criterion in Algorithm QUAD. (Actually, $\|B_{k+1} - B_k\|_F$ is used in the algorithm.)

In the next theorem we examine the block-diagonalization aspect of Algorithm QUAD.

Theorem 3.2. *Let Λ_1 and Λ_2 be the diagonal matrices of the eigenvalues of A inside (a, b) and outside $[a, b]$, respectively, and $B_k\Pi_k = Q_kR_k$ be QR decompositions with column pivoting of B_k. Then, there exist two unitary matrices $U_1 \in \mathcal{C}^{r\times r}$ and $U_2 \in \mathcal{C}^{(n-r)\times(n-r)}$ such that*

$$\|Q_k^H A Q_k - \mathrm{diag}(U_1^H\Lambda_1U_1, U_2^H\Lambda_2U_2)\| \leq \frac{4\|A\|_2}{\sigma_{\min}(R_{11}^{(k)})}\eta_2^{2^k}, \tag{3.3}$$

where $R_{11}^{(k)}$ is the leading principal submatrix of R_k of order r and $\eta_2 = \max_j\{|\alpha(\lambda_j)|^{-1}, \lambda_j \notin [a, b]\}$.

Proof. Let $A = Q\mathrm{diag}(\Lambda_1, \Lambda_2)Q^H$ be the eigenvalue decomposition of A. Partition

$$Q^HQ_k = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}.$$

We can write

$$Q_{11} = U_1 + \Delta U_1, \quad Q_{22} = U_2 + \Delta U_2,$$

where U_i is an optimal unitary matrix approximation of Q_{ii}, $i = 1, 2$. It is not difficult to verify that [7]

$$\|\Delta U_1\|_2 = \max_i|\sigma_i(Q_{11}) - 1| = \|(Q_{11}^HQ_{11})^{-1/2} - I\|_2 = \|(I - Q_{21}^HQ_{21})^{-1/2} - I\|_2 \leq \|Q_{21}\|_2.$$

Similarly, $\|\Delta U_2\|_2 \leq \|Q_{12}\|_2 = \|Q_{21}\|_2$. The last equality holds because Q^HQ_k is unitary. So we have

$$Q^HQ_k\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} + \begin{bmatrix} \Delta U_1 & Q_{12} \\ Q_{21} & \Delta U_2 \end{bmatrix} \equiv U + \Delta U,$$

and

$$\|\Delta U\|_2 \leq \left\|\begin{bmatrix} \Delta U_1 & 0 \\ 0 & \Delta U_2 \end{bmatrix}\right\|_2 + \left\|\begin{bmatrix} 0 & Q_{12} \\ Q_{21} & 0 \end{bmatrix}\right\|_2 \leq 2\|Q_{21}\|_2.$$

Substituting the splitting above into $Q_k^HAQ_k = Q_k^HQ\Lambda Q^HQ_k$ gives

$$Q_k^HAQ_k = U^H\Lambda U + U^H\Lambda\Delta U + \Delta U^H\Lambda Q^HQ_k.$$

Therefore,

$$\|Q_k^H A Q_k - U^H \Lambda U\|_2 = \|U^H \Lambda \Delta U + \Delta U^H \Lambda Q^H Q_k\|_2 \le 2\|A\|_2 \|\Delta U\|_2 \le 4\|A\|_2 \|Q_{21}\|_2. \quad (3.4)$$

To estimate $\|Q_{21}\|_2$, we use Proposition 2.1 again,

$$Q^H Q_k R_k = Q^H B_k \Pi_k = Q^H (I + C^{2^k})^{-1} \Pi_k = (I + D^{2^k})^{-1} Q^H \Pi_k.$$

Partition $D = \operatorname{diag}(D_1, D_2)$ and $R_k = \begin{bmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ 0 & R_{22}^{(k)} \end{bmatrix}$ with $R_{11} \in \mathcal{C}^{r\times r}$. The equality for the $(2,1)$-block reads

$$Q_{21} R_{11}^{(k)} = (I + D_2^{2^k})^{-1} (Q^H \Pi_k)_{21}.$$

Hence $\|Q_{21} R_{11}^{(k)}\| \le \|(I + D_2^{2^k})^{-1}\| < \eta_2^{2^k}$, where $(Q^H \Pi_k)_{21}$ denotes the (2,1)-block of $Q^H \Pi_k$. It follows that

$$\|Q_{21}\|_2 \le \|Q_{21} R_{11}^{(k)}\| \|(R_{11}^{(k)})^{-1}\| < \eta_2^{2^k} / \sigma_{\min}(R_{11}^{(k)}). \quad (3.5)$$

Substituting (3.5) into (3.4), we obtain the bound (3.3).

REMARK. The minimal singular value $\sigma_{\min}(R_{11})$ depends on the particular pivoting strategy used in computing the QR decomposition of B_k. Instead of using QR decomposition with column pivoting, we may also use some of the variants of the so-called rank-revealing QR decompositions or even the more general two-sided orthogonal decompositions.

The results given in the following theorem analyze the structures of the matrices $R^{(k)}$ and M_k. In particular we can show that both of them converge to the identity matrix quadratically.

Theorem 3.3. *Let $R^{(k)}$ be the upper triangular matrix in Algorithm* QUAD *and M_k be the unitary matrix in Proposition 2.1. Then, using the notation of Theorem 3.2, we have*

$$M_k R^{(k)} = I + Q E_k Q^H,$$

where $E_k = \operatorname{diag}(\gamma_1^{(k)}, \ldots, \gamma_n^{(k)})$ with $|\gamma_i^{(k)}| \le \eta^{2^k}$. Furthermore, if all the diagonal elements of $R^{(k)}$ are nonnegative, we have

$$\|R^{(k)} - I\|_2 \le (4 \log_2 n + 8)\eta^{2^k},$$
$$\|M_k - I\|_2 \le (4 \log_2 n + 9)\eta^{2^k}.$$

Proof. Algorithm QUAD gives $B_k = Q_1^{(k)} R^{(k)}$. Substituting the representations of $Q_1^{(k)}$ and B_k given in Proposition 2.1 into the equality above and using the decomposition of $C = QDQ^H$, we have that

$$Q(I + D^{2^k})^{-1} Q^H = Q(I + D^{2^{k+1}})^{-1/2} Q^H M_k R^{(k)},$$

which implies that

$$M_k R^{(k)} = Q(I + D^{2^{k+1}})^{1/2} (I + D^{2^k})^{-1} Q^H \equiv Q(I + E_k) Q^H I + Q E_k Q^H, \quad (3.6)$$

where $E_k = (I + D^{2^{k+1}})^{1/2} (I + D^{2^k})^{-1} - I = \operatorname{diag}(\gamma_1^{(k)}, \ldots, \gamma_n^{(k)})$ and

$$\gamma_i^{(k)} = \frac{\sqrt{1 + d_i^{2^{k+1}}}}{1 + d_i^{2^k}} - 1 = -\frac{2 d_i^{2^k}}{(1 + d_i^{2^k})(1 + d_i^{2^k} + \sqrt{1 + d_i^{2^{k+1}}})}.$$

It is easy to see that for $i \le r$, $|\gamma_i^{(k)}| \le d_i^{2^k} \le \eta^{2^k}$. For $i > r$, we can write

$$\gamma_i^{(k)} = -\frac{2d_i^{-2^k}}{(1+d_i^{-2^k})(1+d_i^{-2^k}+\sqrt{1+d_i^{-2^{k+1}}})},$$

giving $|\gamma_i^{(k)}| \le d_i^{-2^k} \le \eta^{2^k}$. Hence we have that $\|E_k\|_2 \le \eta^{2^k}$.

Now let $\Delta_k = 2QE_k(I+E_k)Q^H$. The representation (3.6) gives

$$(R^{(k)})^H R^{(k)} = Q(I+E_k)^2Q^H = Q(I+2E_k(I+E_k))Q^H = I+\Delta_k,$$

the Cholesky decomposition of $I+\Delta_k$, a perturbation of the identity matrix. By the perturbation theory derived in [5], we get the error bound

$$\|R^{(k)} - I\|_2 \le (2\log_2 n + 4)\|\Delta_k\|_2 \le 4(\log_2 n + 2)\eta^{2^k}.$$

Here we have used the bound $\|\Delta_k\|_2 \le \eta^{2^k}$ that can be easily obtained. Finally, by (3.6),

$$\|M_k - I\|_2 = \|M_k^H - I\|_2 = \|R^{(k)} - I + QE_kQ^HM_k^H\| \le 4(\log_2 n + 2)\eta^{2^k} + \eta^{2^k},$$

completing the proof.

As a consequence of Theorem 3.3, the following corollary shows that $Q_1^{(k)}$ can also be an approximation of the orthogonal projection on to the invariant subspace. The proof is similar and hence is omitted.

Corollary 3.1. *Using the notation in Algorithm* QUAD*, we have*

$$\|Q_1^{(k)} - P_{(a,b)}\|_2 \le \omega_k + (4\log_2 n + 9)\eta^{2^k},$$

where $\omega_k = \max\{\frac{1}{2}|\alpha(\lambda_i)|^{2^{k+1}},\ |\alpha(\lambda_j)|^{-2^k} \mid i \le r < j\} < 1$.

REMARK. The block diagonalizing unitary matrix Q in the last step of Algorithm QUAD does not have to come from the matrix B_p. Corollary 3.1 shows that it can be obtained by the QR decompositions with column pivoting of $Q_1^{(k)}$. We omit the further analysis.

4. Acceleration

The above discussion opens up many possible avenues for accelerating Algorithm QUAD, and we discuss several possibilities in this section. First we can replace the two matrices B_k and $I - B_k$ in Step 2.1 by their powers, i.e., Step 2.1. becomes

$$\begin{bmatrix} B_k^m \\ (I-B_k)^m \end{bmatrix} \begin{bmatrix} Q_1^{(k)} \\ Q_2^{(k)} \end{bmatrix} R^{(k)},$$

and Step 2.2. becomes $B_{k+1} = Q_1^{(k)}(Q_1^{(k)})^H$. This gives other rational iterations for $\{B_k\}$ as

$$B_{k+1} = \phi_m(B_k), \quad k = 1, 2, \ldots,$$

with $\phi_m(x) = \frac{x^{2m}}{x^{2m}+(1-x)^{2m}}$ that has the properties $\phi_m([0,1/2]) = [0,1/2]$ and $\phi_m((1/2,1]) = (1/2,1]$. An interesting observation is the following nested iteration relation among those rational functions:

$$\phi_2 = \phi \circ \phi = \phi^{(2)}, \quad \phi_{2^s} = \underbrace{\phi \circ \cdots \circ \phi}_{(s+1)\text{ times}} = \phi^{(s+1)}(x).$$

It is easy to see that the larger m is the faster the convergence.[3] However, we should also pay attention to the possibility that B_k^m may overflow for large m. On the other hand forming B_k^m and $(I - B_k)^m$ involves extra matrix-matrix multiplications. The best compromise seems to be for $m = 2$ where we can get away with just *one* extra matrix-matrix multiplication and two matrix additions. This is because we can write $(I - B_k)^2 = I - 2B_k + B_k^2$. We summarize the above in the following algorithm.

Algorithm QUART.

1. *Initialization.* $B_0 = \frac{b-a}{2} I$, $Z_0 = A - \frac{a+b}{2} I$.
2. For $k = 0, 1, 2, \ldots$, until convergence
   - 2.1 Compute QR decomposition $\begin{bmatrix} B_k \\ Z_k \end{bmatrix} \begin{bmatrix} Q_1^{(k)} \\ Q_2^{(k)} \end{bmatrix} R^{(k)}$.
   - 2.2 Update $Y_k = Q_1^{(k)} (Q_1^{(k)})^H$, $B_{k+1} = Y_k^2$, and $Z_{k+1} = I - 2Y_k + B_{k+1}$.
   - 2.3 If $\|B_{k+1} - B_k\|_F \leq \texttt{tol}$, then set $p = k + 1$ and terminate.
3. Compute the column pivoting QR decomposition of $B_p \Pi = QR$.
4. Compute $Q^H A Q$ as $Q^H A Q = \begin{bmatrix} \hat{A}_{11} & E_{12} \\ E_{21} & \hat{A}_{22} \end{bmatrix}$, with $\hat{A}_{11} \in \mathcal{C}^{r \times r}$, $r = \text{rank}(R)$.

Another possibility for devising acceleration schemes is to use $\begin{bmatrix} p(B_k) \\ q(I - B_k) \end{bmatrix} = \begin{bmatrix} Q_1^{(k)} \\ Q_2^{(k)} \end{bmatrix} R^{(k)}$, and $B_{k+1} = Q_1^{(k)} (Q_1^{(k)})^H$, where $p(x)$ and $q(x)$ are two polynomials. This gives the rational transformation of B_k as

$$B_{k+1} = \psi(B_k)$$

with $\psi(x) = \frac{p(x)^2}{p(x)^2 + (q(1-x))^2}$. We can tailor the polynomials p and q to accommodate the computation of eigenvalues in other regions in the real line.

5. Comparisons

5.1 Comparison with ISDA

As we mentioned, ISDA uses the incomplete Beta function $\beta_1(x) = x^2(3-2x)$ as the iteration function [1, 3, 10]. Two matrix-matrix multiplications are needed in each iteration. The Beta function $\beta_2(x) = 10x^3 - 15x^4 + 6x^5$ is suggested for acceleration [1, 3, 10]. It requires at least three matrix-matrix multiplications for computing $\beta_2(B)$.

In this section we compare several properties of the functions $\phi = \phi_1$ or ϕ_2 with β_1 and β_2 used as the iteration functions in ISDA. Notice that the eigenvalues of A will be first transformed to the interval $[0, 1]$ by the scaling transformation $B = \ell(A)$.[4] The convergence of iteration

[3] The convergence rate of ϕ_m is $2m$ now.

[4] As a side issue, this step can be accomplished using the QR decomposition of $[I, A]^H$ instead of explicitly estimating the largest and smallest eigenvalues of A using some version of the Geeshgorin Circle Theorem and resorting to a linear transformation as is done in [1, 3].

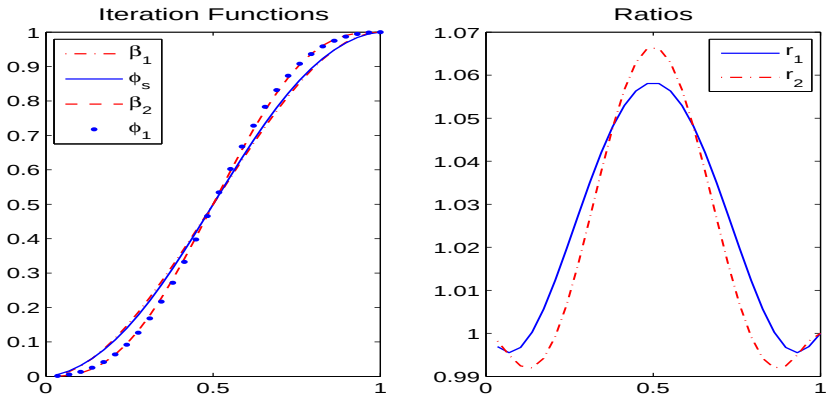

Fig. 5.1. (left) The functions $\beta_1 \approx \phi_s$ with $s = \frac{1}{2}\sqrt[3]{4}$ and $\beta_2 \approx \phi_2$. (right) The rations $r_1 = (\phi_s - \frac{1}{2})/(\beta_1 - \frac{1}{2})$ and $r_2 = (\phi_1 - \frac{1}{2})/(\beta_2 - \frac{1}{2})$.

$\hat{B} = \phi_i(B)$ or $\hat{B} = \beta_i(B)$ depends on the distance $\min_i |\lambda_i(B) - \frac{1}{2}|$ of the eigenvalues of B to the middle point $\frac{1}{2}$ of the interval $[0, 1]$. The larger the distance is, the faster the iteration converges. $\phi_1(x)$ seems to have a slight edge over $\beta_1(x)$ around $1/2$ since

$$\frac{\phi_1(x) - \frac{1}{2}}{\beta_1(x) - \frac{1}{2}} = \frac{4}{3} - \frac{32}{4}(x - \frac{1}{2})^2 + o((x - \frac{1}{2})^2),$$

implying that $\phi_1(x)$ expels away any point close to $1/2$ about $4/3$ as fast as $\beta_1(x)$.

We now compare the function $\phi_2(x)$ and the iterated version of $\beta_1(x)$ and found that for a not too small $x \in (0, 1]$,

$$\phi_2(x) \approx (\beta_1 \circ \beta_1 \circ \beta_1)(x) \equiv \beta_1^{(3)}(x).$$

This implies that β_1 is approximately equal to ϕ_s with $s = \frac{1}{2}\sqrt[3]{4} \approx 0.7937$ and the convergence rate of ISDA using β_1 is about $\sqrt[3]{4} \approx 1.5874$. The accelerated iteration of ISDA using β_2 seems to be quadratically convergent, because $\beta_2(x) \approx \phi_1(x)$ for not small x in (0, 1). Figure 5.1 plots the graphs of ϕ_1, ϕ_s with $s = \frac{1}{2}\sqrt[3]{4}$, β_1, and ϕ_2 on the left and the ratios $r_1 = (\phi_s - \frac{1}{2})/(\beta_1 - \frac{1}{2})$ and $r_2 = (\phi_1 - \frac{1}{2})/(\beta_2 - \frac{1}{2})$ on the right. For numerical examples, see Example 2 and Example 3 below. For our test cases we can conclude that Algorithm QUART is about three times as fast as ISDA and also gives comparable or better accuracy for the computed invariant subspaces.

5.2 Comparison with Algorithm

In this section we compare the accuracy and convergence behavior of the two algorithms: Algorithm 5.2 and Algorithm QUAD. For a detailed discussion of Algorithm CUBIC, the reader is referred to [15]. The main difference between the two algorithms is in Step 2, where the matrices B_k and Z_k are constructed. For Algorithm CUBIC, Step 2 has the following form:[5)]

2.1 Compute QR decomposition $\begin{bmatrix} W_k \\ Z_k \end{bmatrix} \begin{bmatrix} Q_1^{(k)} \\ Q_2^{(k)} \end{bmatrix} R^{(k)}$.

2.2 Update $W_{k+1} = Q_1^{(k)} W_k (Q_1^{(k)})^H$, $\quad Z_{k+1} = Q_2^{(k)} Z_k (Q_2^{(k)})^H$.

[5)] Actually, Algorithm CUBIC used a different but equivalent form as is presented here. The form presented here is used to make the comparison with Algorithm QUAD easier.

The initial values are the same as in Algorithm QUAD. It requires 4 matrix-matrix multiplications. The following table shows the essential characteristics of the algorithms.[6)]

Algorithms	Storage	Matrix-matrix	QR	Convergence rate
QUAD	$3n^2$	1	1	2
CUBIC	$5n^2$	4	1	3
QUART	$3n^2$	2	1	4
ISDA (β_1)	$3n^2$	2	0	≈ 1.59
ISDA (β_2)	$4n^2$	3	0	≈ 2

6. Numerical Experiments

In this section, we show some numerical results to illustrate the efficiency of our proposed algorithms for computing an invariant subspace $\mathcal{V}_{(a,b)}$ for the given interval (a, b). We use synthetic test matrices.

EXAMPLE 1. We construct the test matrix A as follows (in MATLAB notation).

```
n = 500;
[Q,temp] = qr(1-2*rand(n));
d = sort(1-2*rand(n,1));
A = Q*diag(d)*Q';
```

We compute the invariant subspace corresponding to the right half-eigenvalues, i.e., setting $a = \frac{1}{n}\mathrm{tr}(A)$ and $b = \Omega$ where Ω is the estimated bound of the eigenvalues of A using Geeshgorin Circle Theorem. Notice that ISDA is convenient for computing an invariant subspace corresponding to extreme eigenvalues. As we have shown in the last section, the iteration number of ISDA with β_1 is about three times of that for Algorithm QUART, while ISDA with β_2 and Algorithm QUAD has almost the same number of iterations. However, since ISDA requires additional two matrix-matrix multipliers than QUAD and two matrix-matrix multiplications cost much more than on QR decomposition, ISDA with β_2 is generally slower than QUAD. In the following table, we list the experimental results, where err_1, err_2, and err_3 denote the block-diagonalizing error, the approximate error of the computed eigenvalues in (a, b), and the distance between the compute invariant subspace $\mathrm{span}(\hat{Q}_1)$ and the true one $\mathcal{V}_{(a,b)}\mathrm{span}(Q_1)$, respectively,

$$err_1 = \|\hat{Q}_2^T A \hat{Q}_1\|, \quad err_2 = \|\lambda_{(a,b)}(A) - \lambda(\hat{Q}_1^T A \hat{Q}_1)\|, \quad err_3 = \|\hat{Q}_1 - Q_1\|.$$

$n = 500$	CPU time (s)	iter	err1	err2	err3	Flops/n^3
QUAD	19.89	19	1.09e-14	2.10e-14	3.06e-14	194
QUART	14.61	11	9.50e-15	2.31e-14	2.84e-14	135
ISDA (β_1)	20.02	30	1.84e-14	2.39e-14	4.18e-14	183
ISDA (β_2)	21.92	20	5.77e-14	2.17e-14	5.08e-14	203

EXAMPLE 2. In this test, we show the efficiency of our methods for computing an invariant subspace associate the middle eigenvalues of A. We set

$$a = \frac{1}{2}(\lambda_{100} + \lambda_{101}), \quad b = \frac{1}{2}(\lambda_{200} + \lambda_{201}).$$

[6)] When we count the number of matrix-matrix multiplications and the amount of storage, we do not take into account of the symmetry of some of the quantities. This is mainly because it is still not very clear how to handle symmetry in a parallel environment.

For ISDA, if a linear transformation is used for the initial scaling ℓ, it requires that one applies ISDA twice, one for the invariant subspace $\mathcal{V}_{a,\Omega)}$ of A and the other for the invariant subspace $\mathcal{V}_{a,b)}$ of the truncated matrix $Q_1^T A Q_1$. In the following table, we show the CPU times and iteration numbers, respectively.

$n = 500$	CPU time (s)	iter	err1	err2	err3	Flops/n^3
QUAD	11.75	11	3.56e-15	9.53e-15	2.56e-14	113
QUART	9.08	7	3.58e-15	8.15e-15	2.53e-14	87
ISDA (β_1)	27.73+18.16	27+26	8.57e-15	1.19e-14	9.19e-14	251
ISDA (β_2)	29.98+19.80	18+17	2.00e-14	9.12e-15	8.54e-14	276

Acknowledgments. The work of the first author was supported in part by NSFC project 60372033. The work of the second author was supported in part by NSF grant CCR-9619452.

References

[1] L. Auslander and A. Tsao, On parallelizable eigensolvers, *Adv. Appl. Math.*, **13** (1992), 253-261.

[2] Z. Bai, J. Demmel and M. Gu, Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems, Technical report CSD-94-793, Computer Science Division, University of California at Berkeley, 1994.

[3] C. Bischof, S. Huss-Lederman, X. Sun, A. Tsao and T. Turnbull, Parallel studies of the invariant subspace decomposition approach for banded symmetric matrices, Seventh SIAM Conference on Parallel Processing for Scientific Computing, San Francisco, 1995.

[4] A.Y. Bulgakov and S.K. Godunov, Circular dichotomy of the spectrum of a matrix, *Siberian Math. J.*, **29** (1988), 734-744.

[5] A. Edelman and W. Mascarenhas, On Parlett's matrix norm inequality for the Cholesky decomposition, *Numer. Linear Algebra Appl.*, **2** (1995), 243-250.

[6] S.K. Godunov, Problem of the dichotomy of the spectrum of a matrix, *Siberian Math. J.*, **27** (1986), 649-660.

[7] G.H. Golub and C.F. Van Loan, Matrix Computations, 2nd edition, Johns Hopkins University Press, Baltimore, Maryland, 1989.

[8] N.J. Higham, The matrix sign decomposition and its relation to the polar decomposition, *Linear Algebra Appl.*, **212/213** (1994), 3-20.

[9] J. Howland, The sign matrix and the separation of matrix eigenvalues, *Linear Algebra Appl.*, **49** (1983), 221-232.

[10] S. Huss-Lederman, A. Tsao and T. Turnbull, A parallelizable eigensolver for real diagonalizable matrices with real eigenvalues, *SIAM J. Sci. Comput.*, **18(3)** (1997), 869-885.

[11] A. Malyshev, Computing the invariant subspaces of a regular linear pencil of matrices, *Siberian Math. J.*, **30** (1989), 559-567.

[12] A. Malyshev, Parallel algorithm for solving some spectral problems of linear algebra, *Linear Algebra Appl.*, **188/189** (1993), 489-520.

[13] P. Pandey, C. Kenney and A. Laub, A parallel algorithm for the matrix sign function, *Int. J. High Speed Comput.*, **2** (1990), 181-191.

[14] ScaLAPACK, `http://www.netlib.org/scalapack/index.html`.

[15] H. Zha and Z. Zhang, A cubically convergent parallelizable method for the Hermitian eigenvalue problem, *SIAM J. Matrix Anal. Appl.*, **19** (1998), 468-486.

[16] H. Zha, Z. Zhang and W. Ying, A parallelizable quadratically convergent QR-like method without shifts for Hermitian eigenvalue problem, *Linear Algebra Appl.*, **417** (2006), 478-495.

[17] Z. Zhang and O. Tianwei, Computing eigenvectors of symmetric matrices using simple inverse iterations, *J. Comput. Math.*, **21(5)** (2003), 657-670.